

# Install on Windows



## Missing VC runtime

If you encounter errors during the install or upgrade process that indicate the system can't start the shibd daemon/service process, it's likely that the installer is failing to install the Visual C++ runtime libraries for some reason, and you may have to pre-install those manually. At some point Microsoft is planning to make it impractical for us to redistribute these files with the installer, at which point this workaround is probably going to be universal.

These links may break at some point, but for now the 32-bit and 64-bit runtimes can be found at:

- [https://aka.ms/vs/15/release/VC\\_redist.x86.exe](https://aka.ms/vs/15/release/VC_redist.x86.exe)
- [https://aka.ms/vs/15/release/VC\\_redist.x64.exe](https://aka.ms/vs/15/release/VC_redist.x64.exe)

The top-level link to find them is <https://visualstudio.microsoft.com/downloads/> via Other Tools.

- [Installation](#)
- [Upgrades](#)
- [Web Server Overview](#)
  - [Installation with IIS](#)
  - [Upgrades with IIS](#)
  - [Installation with Apache](#)
  - [Installation with Netscape/Sun/Oracle Java Web Server](#)
  - [FastCGI Unsupported](#)
- [Running the Installer](#)
  - [Shibboleth Service](#)
  - [Failure to Install](#)
- [Source Builds](#)
- [Initial Testing](#)

## Installation

The SP is available for Windows with modules for all the supported web servers. There is an installer available for the supported Windows Server versions, 2008 and above.

Earlier Windows and Service Pack versions are **not** usable as of V3.

Desktop versions of Windows that are of subsequent vintage will generally work but are not formally supported.

The Windows installer contains a fourth version field that indicates the patch level within a particular SP release. Initially 0, it will be incremented if patches to software included with but not part of the SP need to be updated (e.g., OpenSSL). Subsequent patch level installers will safely upgrade older versions and the [ReleaseNotes](#) will always document exactly what library versions are included in each release.

## Upgrades

Upgrading to new releases is handled automatically when the MSI installer is used. The system prevents configuration files from being overwritten and skips "initial install" tasks like generating keys. The Shibboleth daemon is restarted by the package but you will need to restart the web server you're using yourself.

## Web Server Overview



### Restrictive ACL on the configuration file

New Installations place a restrictive file security ACL on the configuration directory. If you run your web server under a different user account (not a member of the `Administrators` group) you will need to adjust this ACL.

## Installation with IIS

The plugin modules in support of IIS are always installed. If IIS itself is installed you will be prompted "Configure IIS7 module?". Check this box to actually configure Shibboleth to run with IIS, in practice this can be done with two command.

See [this topic](#) for details.

In contrast to older versions, no older IIS6 compatibility tools are needed.

## Upgrades with IIS

Existing installations configured to work with IIS will continue to do so, with the older ISAPI plugin updated. See [this topic](#) for more details, particularly to migrating to the newer plugin.

## Installation with Apache

The versions of Apache available from the <http://www.apachelounge.com/> web site are known to work with the modules that come with the SP package with no known restrictions.

Other versions might work, but they also might not work. Versions with significantly altered header files, such as IBM's or Oracle's will definitely not work unless you build the Shibboleth module from source.

Only Apache 2.2 and 2.4 are supported.

For more details on Apache see [this topic](#).

## Installation with Netscape/Sun/Oracle Java Web Server

An NSAPI module continues to be included as a transition tool for anybody still using it, but it is deprecated and we suggest moving off it as soon as possible.

## FastCGI Unsupported

As of V3, we no longer provide modules for FastCGI on Windows due to lack of formal support for the underlying FastCGI library.

## Running the Installer

The installer is run in the usual way and can be run without a UI.

The following properties can be set

Property	Description
KEYGEN_EXTRAS	Allows extra parameters to be passed to the keygen command used during installation. For instance, <pre>msiexec /i shibboleth-sp-3.0.0-win64.msi KEYGEN_EXTRAS="-h newSP.department.university.edu"</pre> allows the addition of a subjectAltName in the generated certificate.
ALWAYS_START_SERVICE	Set this to <code>FALSE</code> to prevent the installer from starting the shibd service. This can be useful to debug installations (see below).

## Shibboleth Service

Once installation is complete, you'll need to run the Shibboleth daemon, `shibd`, at all times that the SP is in use. `shibd` is a console application that is usually installed as a Windows service.

- To run the process in console mode for testing or to diagnose major problems, supply a `-console` parameter when running it.
- If `shibd` won't start, use the `-check` option from the command line to echo most logging information to the console for debugging.

Other parameters can be used to install (or remove) `shibd` from the service database and subsequent control is generally via the Service Control Manager applet.



### Monitoring the Service

Newer versions of Windows support automatic restart of failed services. We suggest using this feature to restart `shibd` when it fails. Although stability is good, maximum reliability will be achieved by monitoring the process.

## Failure to Install

The most common reason for the installation failing is that the Shibboleth service (above) does not start correctly. First, refer to the note at the top of this page and rule that out. If that doesn't bear fruit, in order to debug this you can instruct the installer to **not** try to start the service by specifying that the `ALWAYS_START_SERVICE` property contain the value `FALSE`. Do this from the command line:

### Installing with no service start

```
c:\> msisexec /i Installer.msi ALWAYS_START_SERVICE=FALSE
```

You can then use the `-check` option described above to debug why the service will not start. Usually the problem tends to be a DLL conflict with some existing copy of one of the libraries we ship, but we have generally worked around this risk by renaming all our libraries in ways that tend not to cause conflicts.

Once this is completed you can start the service manually.

### Starting the Shibboleth Service from the command line

```
c:\> sc start shibd_default
```

In some situations attempting the installer may appear to do nothing when double-clicked, and if invoked with `msiexec /i` on the command line may throw the error "This installation package could not be opened. Verify that the package exists and that you can access it, or contact the application vendor to verify that this is a valid Windows Installer package." If this occurs, a viable workaround has found to be to create a new .bat file containing the msi install command listed below, and then drag the downloaded msi onto it.

### Windows batch file to run MSI installer by drag and drop

```
msiexec /i %1
```

## Source Builds

We do not recommend this option, but we have a [description](#) of the process.

## Initial Testing

You can test to ensure that the SP is running properly and the surrounding environment is correct by accessing

```
https://localhost/Shibboleth.sso/Status
```

**from the actual web server machine** itself. You **MUST** use "localhost" as the hostname or it WILL NOT WORK by default. If this test is successful, then the software is ready for further configuration.



You can also access the Status handler from other clients or using a non-localhost name, but **only** if you change the `acl` parameter in the configuration to permit your client address or remove it entirely to open up access to anybody. The ACL is present by default because the Status handler can return some arguably sensitive information about your configuration.

Now you can progress to the [GettingStarted](#) material. Once you've actually configured the SP with its own settings and metadata from at least one IdP, in order to check that the SP is "working":

1. Protect a directory by requiring a Shibboleth session. Often, this is already done by default for the location `/secure`.
2. Next, you typically place a script inside the protected directory that dumps the web server environment. With PHP for example you could in the easiest case just place a script there with the following:

```
<?php print_r($_SERVER) ?>
```

For an IIS deployment, an example ASP.NET script:

```
<% @ Page Language="C#" %>
<%
Response.Write("<h3>Server Variables</h3>");
foreach (string var in Request.ServerVariables)
{
    Response.Write(var + " = " + Request[var] + "<br>");
}
%>
```

3. Make sure that the Shibboleth-supplied variables are present. If there is a non-empty variable called **Shib-Application-ID**, you successfully authenticated and have a valid session. However, you also should check if there are other non-empty Shibboleth variables defined in the **attribute-map.xml** file. If there are no variables like **mail** or **givenName** or **surname**, the IdP probably released no attributes. In either case, have a look at the *shibd.log* and *transaction.log* files.